IN THE CLAIMS:

1. (Currently Amended): A method for exchanging Java class objects between two computing entities in an object-oriented programming environment using a transport mechanism in which said Java class objects are contained in files, each file defining a resource, each resource designed to contain a plurality of particular ones of said objects, said method comprising the steps of:

(1) providing a resource factory for building resources in the form of XMI documents, said factory including a plurality of software modules, each software module adapted for building resources from a data source responsive to a request for ~~an~~ a Java class object of a type to which said resource corresponds, each said software module designed to build a resource of a particular type;

(2) responsive to a request for ~~an~~ a Java class object from a first computing entity, selecting a software module for building a resource of the type to which said Java class object corresponds;

(3) subsequent to step (2), building ~~a resource~~ an XMI document for containing said Java class object using said selected software module, said ~~resource~~ XMI document populated with information defining said ~~resource~~ XMI document, but not containing said Java class object;

(4) subsequent to step (3), inserting said Java class object into said ~~resource~~ XMI document;

(5) subsequent to step (4), transmitting said ~~resource~~ XMI document to said first computing entity using said transport mechanism; and

(6) subsequent to step (5), providing to said first computing entity said Java class object.

2. (Previously Presented): The method of claim 1 wherein, in step (4), only said Java class object is inserted in said ~~resource~~ XMI document.

3. (Previously Presented): The method of claim 2 further comprising the steps of:

(7) providing a reflection adapter factory for populating Java class objects within ~~resources~~ XMI documents, said factory adapted to provide software modules for populating Java class objects, each said software module designed for an environment corresponding to ~~an~~ a Java class object;

(8) responsive to a request for a property of said Java class object, selecting a one of said reflection adapters for the environment of the particular property;

(9) populating said Java class object with said property; and

(10) providing to said first computing unit said property.

4. (Original): The method of claim 3 wherein said Java class object comprises a plurality of properties and step (9) comprises populating said Java class object with all properties of said Java class object that can be reflected.

5. (Currently Amended): A method for exchanging Java class objects between two computing entities in an object-oriented programming environment using a transport mechanism in which said Java class objects are contained in files, each file defining a resource, each resource designed to contain a plurality of particular ones of said Java class objects, said method comprising the steps of:

(1) providing a resource factory for building resources in the form of XMI documents, said factory including a plurality of software modules for building resources from a data source, each said software module designed to build a resource of a particular type;

(2) determining whether said first computing entity has stored a resource containing said Java class object;

(3) if said first computing entity has stored a resource corresponding to said Java class object, determining if said corresponding resource stored at said first computing entity contains said Java class object;

(4) if said corresponding resource stored at said first computing entity does not contain said Java class object, said first computing entity issuing a request for said Java class object;

(5) responsive to a request for said Java class object from said first computing entity, selecting a software module for building a resource of the type to which said Java class object corresponds, said resource being in the form of an XMI document;

(6) subsequent to step (5), building ~~a resource~~ an XMI document for containing said Java class object using said selected software module, said ~~resource~~ XMI

document populated with information defining said resource, but not containing said Java class object;

(7) subsequent to step (6), inserting only said Java class object into said ~~resource~~ XMI document;

(8) subsequent to step (7), transmitting said ~~resource~~ XMI document to said first computing entity using said transport mechanism; and

(9) subsequent to step (8), providing to said first computing entity said Java class object.

6. (Currently Amended): A method for exchanging Java class objects between two computing entities in an object-oriented programming environment using a transport mechanism in which said Java class objects are contained in files, each file defining a resource, each resource designed to contain a plurality of particular ones of said Java class objects, said method comprising the steps of:

(1) providing a resource factory for building resources in the form of XMI documents, said factory including a plurality of software modules for building resources from a data source, each said software module designed to build a resource of a particular type;

(2) responsive to a request for ~~an~~ a Java class object from a first computing entity, selecting a software module for building a resource of the type to which said Java class object corresponds;

(3) subsequent to step (2), building a resource in the form of an XMI document for containing said Java class object using said selected software module, said

resource populated with information defining said resource, but not containing said Java class object;

(4) subsequent to step (3), inserting said Java class object into said ~~resource~~ XMI document;

(5) subsequent to step (4), transmitting said ~~resource~~ XMI document to said first computing entity using said transport mechanism;

(6) subsequent to step (5), providing to said first computing entity said Java class object.

(7) providing a reflection adapter factory for populating Java class objects within resources, said factory adapted to provide software modules for populating Java class objects, each said software module designed for an environment corresponding to an Java class object;

(8) determining whether said first computing entity has stored ~~said~~ a property of said Java class;

(9) if said first computing entity has not stored said property, issuing a request for said property;

(10) responsive to said request for said property of said Java class object, selecting a one of said reflection adapters for the environment of the particular property;

(11) populating said Java class object with said property; and

(12) providing to said first computing unit said property.

7-9. (Cancelled).

10. (Original): The method of claim ~~9~~ 6 wherein steps (4) and (5) utilize the Meta Object Facility of the Object Management Group specification to read an XMI document.

11. (Original): The method of claim ~~8~~ 6 wherein, in step (2), said information defining said resource comprises at least a package object of said resource.

12. (Currently Amended): A method for exchanging Java class objects between two computing entities in an object-oriented programming environment using a transport mechanism in which said Java class objects are contained in XMI documents ~~files~~, each file defining a resource, each resource designed to contain a plurality of particular ones of said Java class objects, said method comprising the steps of:

(1) providing a resource factory for building resources, said factory including a plurality of software modules for building resources from a data source, each said software module designed to build a resource of a particular type;

(2) responsive to a request for an Java class object from a first computing entity, selecting a software module for building ~~a resource~~ an XMI document of the type to which said Java class object corresponds;

(3) subsequent to step (2), building ~~a resource~~ an XMI document for containing said Java class object using said selected software module, said ~~resource~~ XMI

document populated with information defining said resource, but not containing said Java class object;

(4) subsequent to step (3), inserting said Java class object into said ~~resource~~ XMI document;

(5) subsequent to step (4), transmitting said ~~resource~~ XMI document to said first computing entity using said transport mechanism;

(6) subsequent to step (5), providing to said first computing entity said Java class object.

(7) providing a reflection adapter factory for populating Java class objects within resources, said factory adapted to provide software modules for populating Java class objects, each said software module designed for an environment corresponding to ~~an~~ a Java class object;

(8) determining whether said first computing entity has stored ~~said~~ a property of said Java class;

(9) if said first computing entity has not stored said property, issuing a request for said property;

(10) responsive to a request for said property of said Java class object, selecting a one of said reflection adapters for the environment of ~~the particular~~ said property;

(11) determining whether said selected reflection adapter has previously reflected said ~~requested~~ property;

(12) if said first computing entity has previously reflected said ~~requested~~ property, populating said Java class object with said property; and

(13) providing to said first computing unit said property.

13-15. (Cancelled).